Authored by: Indhu Chandramohan

There are 2 types of windows function application:

1. Aggregate function
2. Ranking function

```
--Sample table exploration (employees table)
select * from hr.employees;
```

1. Aggregate function

```
--learn to use simple sum function usage
SELECT SUM(salary) AS total_salary
    FROM hr.employees;
```

```
--learn how to use group by clause
SELECT department_id, SUM(salary) AS department_salary
    FROM hr.employees
    GROUP BY department_id
    ORDER BY department_id;
```

```
--learn to use over() and partition by
-- OVER is used for window set (set of records where the function is applied on)
-- PARTITION BY is used to partition each window set
SELECT
     department_id, job_id, salary,
     SUM(salary) OVER() AS total_salary,
     SUM(salary) OVER(PARTITION BY department_id) AS department_salary
    FROM hr.employees
    ORDER BY department_id, job_id, salary;
```

```
--learn to use row_number function
SELECT
        department_id, job_id, salary,
        ROW_NUMBER() OVER(PARTITION BY department_id ORDER BY job_id)
AS row_num2
      FROM hr.employees;
```

2. Ranking function

```
--learn to use rank function
SELECT department_id,
job_id,
salary,
RANK() OVER(PARTITION BY department_id ORDER BY salary DESC) AS rank
FROM hr.employees;
```

```
--learn to use dense rank function and how it is different from rank function
SELECT department_id,
job_id,
salary,
DENSE_RANK() OVER(PARTITION BY department_id ORDER BY salary DESC) AS
dense_rank
FROM hr.employees;
```

Explanation: Difference between rank() Vs dense_rank()

1. RANK skips positions after equal rankings, but DENSE_RANK don’t.

2. The number of positions skipped depends on how many rows had an identical ranking.

   For example, Mohan and Byju sold the same number of products and are both ranked as #2. With RANK , the next position is #4;
   With DENSE_RANK , the next position is #3.